CLAIMS

What is claimed is:

1. A computer-implemented method for optimizing an executable program having a plurality of functions and at least one function with a first name associated with executable code that implements the function at a first address and at least one linkage stub code segment having code that branches to the first address and a symbolic name by which the function is invoked in the program, comprising:

identifying branch instructions having target addresses that reference the linkage stub code segment; and

replacing the target addresses of the branch instructions with the first address.

2. The method of claim 1, further comprising replacing the target address of the branch instructions with the first address only in functions that are reached during program execution.

3. The method of claim 1, further comprising:

searching a symbol table for an entry having a symbolic name that that is a derivation of the first name and reading a linkage stub address associated with the symbolic name; and

replacing target addresses of branch instructions having target addresses equal to the linkage stub address with an address at which the code that implements the function is stored.

4. The method of claim 1, further comprising:

searching a symbol table for an entry having a symbolic name that matches the first name with an underscore prefix and reading a linkage stub address associated with the symbolic name; and

replacing target addresses of branch instructions having target addresses equal to the linkage stub address with an address at which the code that implements the function is stored.

5. The method of claim 1, further comprising:

searching a symbol table for an entry having a symbolic name that matches the first name with an underscore suffix and reading a linkage stub address associated with the symbolic name; and

replacing target addresses of branch instructions having target addresses equal to the linkage stub address with an address at which the code that implements the function is stored.

6. The method of claim 1, further comprising:

replacing function entry points in the executable program with breakpoints, whereby breakpointed functions are generated; and

upon encountering a breakpoint of a breakpointed function during program execution, identifying within the breakpointed function branch instructions that target linkage stub functions.

7. The method of claim 6, further comprising:

storing original instructions from the function entry points prior to replacement with the breakpoints;

upon encountering a breakpoint of a breakpointed function during program execution, restoring the original instruction to the entry point of the breakpointed function.

8. The method of claim 6, further comprising:

searching a symbol table for an entry having a symbolic name that that is a derivation of the first name and reading a linkage stub address associated with the symbolic name; and

replacing target addresses of branch instructions having target addresses equal to the linkage stub address with an address at which the code that implements the function is stored.

9. The method of claim 6, further comprising:

searching a symbol table for an entry having a symbolic name that matches the first name with an underscore prefix and reading a linkage stub address associated with the symbolic name; and

replacing target addresses of branch instructions having target addresses equal to the linkage stub address with an address at which the code that implements the function is stored.

10. The method of claim 6, further comprising:

searching a symbol table for an entry having a symbolic name that matches the first name with an underscore suffix and reading a linkage stub address associated with the symbolic name; and replacing target addresses of branch instructions having target addresses equal to the linkage stub address with an address at which the code that implements the function is stored.

11. The method of claim 1, further comprising:

replacing entry points of linkage stub code segments in the executable program with breakpoints, whereby breakpointed linkage stubs are generated; and

upon encountering a breakpoint of a breakpointed linkage stub during program execution, changing a target address of a branch instruction that branched to the breakpointed linkage stub to reference the function referenced by the breakpointed linkage stub.

12. The method of claim 11, further comprising:

searching a symbol table for an entry having a symbolic name that that is a derivation of the first name and reading a linkage stub address associated with the symbolic name; and

replacing target addresses of branch instructions having target addresses equal to the linkage stub address with an address at which the code that implements the function is stored.

13. The method of claim 11, further comprising:

searching a symbol table for an entry having a symbolic name that matches the first name with an underscore prefix and reading a linkage stub address associated with the symbolic name; and replacing target addresses of branch instructions having target addresses equal to the linkage stub address with an address at which the code that implements the function is stored.

14. The method of claim 11, further comprising:

searching a symbol table for an entry having a symbolic name that matches the first name with an underscore suffix and reading a linkage stub address associated with the symbolic name; and

replacing target addresses of branch instructions having target addresses equal to the linkage stub address with an address at which the code that implements the function is stored.

15. An apparatus for optimizing an executable program having a plurality of functions and at least one function with a first name associated with executable code that implements the function at a first address and at least one linkage stub code segment having code that branches to the first address and a symbolic name by which the function is invoked in the program, comprising:

means for identifying branch instructions having target addresses that reference the linkage stub code segment; and

means for replacing the target addresses of the branch instructions with the first address.